基于时空信息和任务流行度分析的移动群智感知任务推荐 *

杨桂松 [1a]，王静茹 [1a]，李　俊 [2]，何杏宇 [1b†]

(1. 上海理工大学 a. 光电信息与计算机工程学院; b. 出版印刷与艺术设计学院, 上海 200093; 2. 国家工业信息安全发展研究中心, 北京 100040)

摘　要：现有移动群智感知任务推荐的共同缺点是：一方面，未充分考虑时空信息对工人偏好的影响，导致推荐准确性低；另一方面，忽略任务流行度对推荐的影响，导致推荐覆盖率差。为解决这些问题，本文提出一种基于时空信息和任务流行度分析的移动群智感知任务推荐方法。首先，充分利用工人执行记录中的相关信息(如工人执行任务的时间、位置)，准确预测工人对任务的偏好。其次，基于工人声誉和任务执行情况分析任务流行度并设计任务流行度惩罚因子，提升推荐效果的覆盖率。然后，结合工人偏好和流行度惩罚因子生成任务推荐列表。实验结果表明，与现有基线方法相比，所提出方法在推荐准确率上平均提升了 3.5%，推荐覆盖率上平均提高了 25%。

关键词：移动群智感知；任务推荐；时空信息；流行度偏差；任务流行度

中图分类号：TP393　　doi: 10.19734/j.issn.1001-3695.2022.01.0046

Task recommendation based on spatial-temporal information and task popularity analysis in mobile crowd sensing

Yang Guisong[1a], Wang Jingru[1a], Li Jun[2], He Xingyu[1b†]

(1. *a. School of Optical-Electrical & Computer Engineering*, *b. College of Communication & Art Design*, *University of Shanghai for Science & Technology*, *Shanghai* 200093, *China*; 2. *National Industrial Information Security Development Research Center*, *Beijing* 100040, *China*)

Abstract: The drawbacks of existing task recommendation in mobile crowd sensing were as follows: on the one hand, not fully considering the influence of spatial-temporal information on worker preference led to low accuracy of recommendation; On the other hand, ignoring the impact of task popularity on recommendation led to poor recommendation coverage. To solve these drawbacks, this paper proposed a novel task recommendation approach based on spatial-temporal information and task popularity analysis in mobile crowd sensing. Firstly, this approach made full use of the relevant information contained in the worker execution record (e. g. , the time and location of worker performing tasks) to accurately predict the preference of worker for performing tasks. Secondly, in order to reduce the impact of popular tasks on recommendation coverage, this paper analyzed task popularity based on worker reputation and task execution record, and designed appropriate task popularity penalty factor. Then, combining worker preference and task popularity penalty factor, this paper provided an appropriate task recommendation list for each worker. Finally, the experimental results show that compared with the existing baseline methods, the proposed method improves the recommendation accuracy by 3.5% and the recommendation coverage by 25%.

Key words: mobile crowd sensing; task recommendation; spatial-temporal information; popularity bias; task popularity

0　引言

移动群智感知(Mobile crowd sensing, MCS)[1]是一种具有广阔前景的感知模式，已广泛应用于很多领域，例如环境监测[2]、健康数据收集[3]、工业控制管理[4]、地质勘测[5]和交通规划[6]等。典型的移动群智感知系统由任务发布方、任务完成方(工人)和 MCS 平台三部分组成。其工作流程如下：任务发布方将任务请求发布在 MCS 平台，MCS 平台根据接收到的任务请求为工人建立任务列表，工人从列表中选择任务去执行，并将感知数据上传至 MCS 平台，MCS 平台最终将感知数据反馈给任务发布方。在这个过程中，工人选择任务的可能性受工人偏好的影响。因此， 如何挖掘工人偏好，向工人推荐最有可能执行的任务列表尤为重要。任务推荐的效率和质量不仅会影响工人的执行意愿，也会影响任务的感知质量。

现有研究倾向于使用单一模型来学习工人偏好[7]，进而实现任务推荐。例如，基于工人相似性或者任务相似性[8~10]的任务推荐方法、基于逻辑回归模型的方法[11, 12]。这些方法都需要提前制定影响工人偏好的因素，但是在移动群智感知中，影响工人偏好的因素众多，很难提前制定所有因素。为解决这个问题，一些研究开始将推荐系统[13]应用到移动群智感知领域的任务推荐，从工人的历史执行记录中预测工人偏好。例如，文献[14, 15]利用矩阵分解模型预测工人偏好，并依据偏好为工人生成任务推荐列表。文献[16]提出一种基于特征的贝叶斯任务推荐方法，通过任务类别来学习任务的潜在特征，解决新任务冷启动情况下的任务推荐问题。文献[17]综合考虑每个工人的偏好和可靠性实现任务推荐。

上述方法未充分考虑工人偏好的时空变化，这会影响推荐准确度。一方面，工人的偏好随着时间动态变化，工人近期的任务选择更能代表工人目前的偏好，另一方面，移动群智感知中的任务分布在不同的感知区域，工人需要移动到感

收稿日期：2022-01-22；**修回日期：**2022-03-25　**基金项目：**国家自然科学基金资助项目(61602305，61802257)；上海市自然科学基金资助项目(18ZR1426000，19ZR1477600)

作者简介：杨桂松(1982-)，男，河南漯河人，副教授，硕导，博士，主要研究方向为物联网与普适计算等；王静茹(1995-)，女，河南开封人，硕士研究生，主要研究方向为移动群智感知；李俊(1986-)，男，江西吉安人，正高级工程师，博士，主要研究方向为工业领域网络与数据安全；何杏宇(1984-)，女(通信作者)，湖南岳阳人，副教授，硕导，博士，主要研究方向为物联网、群智计算和大数据分析(xy_he@usst.edu.cn)．

知区域地点去执行任务，出于自身成本角度考虑，工人更倾向于执行距离自己较近位置的任务。尽管文献[18]利用张量分解模型实现基于工人位置的任务推荐，但是受到工人流动性的影响[19]，该模型需要频繁更新来适应工人的位置变化，导致不能及时捕捉工人新的偏好。因此，如何结合工人时空信息来准确预测工人偏好是一大挑战。

另外，现有移动群智感知任务推荐研究忽略了任务流行度对推荐效果的影响，这会产生流行度偏差问题[20]，造成任务推荐覆盖率下降。例如，市区的交通流量监测等流行任务可能会因为其执行简单而受到大量工人喜欢，如果平台为这些工人都推荐该任务，会产生感知数据冗余，而诸如山区地质水平测量等冷门任务则可能因为没得到足够的推广而收集不到充足的样本导致任务失败。一般情况下，推荐系统领域中任务的流行程度由执行该任务的工人比例决定，比例越高，则任务流行度越高[21]。在移动群智感知系统中，工人具有从众心理，容易受到其他工人的影响而去执行某个任务，导致任务的流行度发生变化。因此，任务流行度和工人声誉关系密切，声誉越高的工人对其他工人的影响更大，进而对任务流行度的影响也就更大。因此，基于工人声誉和任务执行情况分析任务的流行度，设计合理的流行度惩罚因子，提高任务推荐覆盖率，是另一个挑战。

针对上述挑战，本文提出基于时空信息和流行度分析的移动群智感知任务推荐方法(Time Matrix Factorization and Bidirectional Long Short-Term Memory, TimeMF-BiLSTM)，该方法综合考虑工人偏好和任务流行度惩罚因子为工人推荐任务列表，保证任务推荐的准确度和覆盖率。本文的主要贡献如下：

a) 使用融合时间因素的矩阵分解模型从工人历史执行数据中学习工人和任务的隐含关系，再利用双向长短期记忆网络模型从工人的轨迹信息中提取工人的位置偏好信息，将两个模型得到的偏好融合得到精确度更高的任务推荐。

b) 引入社交网络计算工人声誉，提出一种基于工人声誉信息的任务流行度计算方法，实现对任务流行度更精确分析，并设计合理的任务流行度惩罚因子，保证任务推荐的覆盖率。

1　系统模型

图 1 展示了 TimeMF-BiLSTM 任务推荐方法的整体架构。该方法先使用融合时间因素的矩阵分解模型(Time Matrix Factorization, TimeMF)从工人执行记录中学习工人和任务的隐含关系(即工人潜在特征向量和任务潜在特征)，得出工人对任务的偏好(记为 TimeMF 偏好评分)，解决工人偏好随时间动态改变的问题；再使用双向长短期记忆网络(Bidirectional Long Short-Term Memory, BiLSTM)从工人 GPS 轨迹序列中学习其下一时刻可能到达的位置，并基于任务与该位置之间的距离得到工人偏好(记为 BiLSTM 偏好评分)，解决工人偏好随着工人位置动态改变的问题；然后，将 TimeMF 模型学习到的工人偏好和 BiLSTM 模型学习到的工人偏好使用逻辑回归方法融合在一起，综合得到工人的偏好评分，保证任务推荐的准确性；此外，引入社交网络计算工人声誉，并根据任务执行情况分析工人之间的影响因子，结合工人声誉和工人影响因子计算任务流行度，构造与任务流行度负相关的惩罚因子；使用流行度惩罚因子和工人偏好相乘得到工人对任务的最终评分，解决由于任务流行度偏差造成的感知数据覆盖不平衡问题。在此基础上，将工人对任务的最终评分从高到低进行排序，为每一个工人生成 *TopN* 任务推荐列表。

本文使用的主要参数描述如下：m 个工人的集合表示为 $\mathbb{U}=\{u_1,u_2,\ldots,u_m\}$，$n$ 个任务的集合表示为 $\mathbb{V}=\{v_1,v_2,\ldots,v_n\}$，每个工人的执行记录表示为 $\mathbb{H}_{u_i}$，$\mathbb{H}_{u_i}$ 包含工人 u_i 执行的所有任务，每个任务的执行情况表示为 $\mathbb{F}_{v_j}$，$\mathbb{F}_{v_j}$ 包括执行任务 v_j 的所有工人 ID 编号，$\mathbb{L}_{u_i}=\{(t_1,l_1),(t_2,l_2),\ldots,(t_s,l_s),\ldots\}$ 表示工人 u_i 的 GPS 轨迹序列，l_s 是由经度和纬度组成的二元组，代表工人 u_i 在时刻 t_s 时的位置。本文提出一种有效的方法来精确计算工人和任务之间的评分矩阵 $\boldsymbol{Y}$，为每个工人生成 *TopN* 任务推荐列表，目标是最大化任务推荐的准确率和覆盖率，任务推荐列表的长度 N 由平台根据实际情况设置。

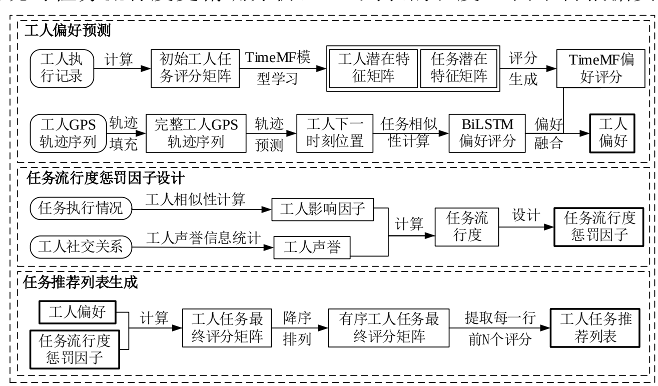


图 1　基于时空信息和流行度分析的移动群智感知任务推荐

Fig. 1　Task recommendation based on spatial-temporal information and popularity analysis in mobile crowd sensing

2　工人偏好预测

2.1　融入时间因素的矩阵分解

为解决工人偏好随时间动态改变的问题，使用 TimeMF 模型学习工人偏好。TimeMF 模型通过不断迭代学习工人潜在特征向量 $\boldsymbol{U}$ 和任务潜在特征向量 $\boldsymbol{V}$，并利用两者乘积预测工人任务评分矩阵 $\boldsymbol{MP}$，即将原始工人任务评分矩阵 $\boldsymbol{MP}$ 中的零值补全。TimeMF 模型主要包含三个部分：a)初始工人任务评分矩阵构建。根据工人执行记录构建原始工人任务评分矩阵，并考虑时间因素对工人偏好的影响；b)TimeMF 模型学习。根据原始工人任务评分矩阵，采用合适的算法对任务潜在特征进行迭代更新，并最终达到一个最优解，使得工人潜在特征向量和任务潜在特征向量的乘积逼近原始工人任务评分矩阵；c)偏好评分生成。根据上一阶段最终得到的工人潜在特征向量和任务潜在特征向量，实现工人对任务的评分预测。

2.1.1　初始工人任务评分矩阵构建

工人对任务的评分表示工人对该任务的偏好，由于 MCS 平台中很少要求工人对任务有显式的评分，因此本文使用工人执行任务的次数来代替工人对任务的评分，工人 u_i 执行任务 v_j 以用三元组 $\{u_i,v_j,t_{ij}\}$ 来表示，t_{ij} 是工人 u_i 执行任务 v_j 的时刻。若工人 u_i 时刻 t_{ij} 执行任务 v_j，则表示工人 u_i 对任务 v_j 感兴趣，$\{u_i,v_j,t_{ij}\}=1$，反之，则表明该时刻工人 u_i 对任务 v_j 不感兴趣，$\{u_i,v_j,t_{ij}\}=0$。

考虑到工人偏好的时间动态特性，工人最近一段时间执行的任务更能代表工人的偏好，因此，引入牛顿冷却定律的思想，通过指数衰减函数 $e^{-\lambda(t-t_{ij})}$ 使评分信息的权重随时间呈

指数衰减。其中，t 是当前时刻，t_{ij} 是工人 u_i 执行任务 v_j 的时刻，λ 是时间衰减因子，$\lambda>0$，且 λ 越大，历史偏好的重要程度越低。在此基础上，建立工人任务原始评分矩阵 $\boldsymbol{MP}$，$\boldsymbol{MP}$ 中的每一个子元素 mp_{ij} 通过下式计算：

$$mp_{ij}=\sum\nolimits_{v_j\in\mathbb{H}_{u_i}} e^{-\lambda(t-t_{ij})}(u_i,v_j) \tag{1}$$

2.1.2 TimeMF 模型学习

工人潜在特征向量和任务潜在特征向量用两个 k 维的低秩特征矩阵 $\boldsymbol{U}$ 和 $\boldsymbol{V}$ 表示，$U\in\mathbb{R}^{m\times k}$ 的每一行代表一个工人 u_i 的潜在特征向量 U_i，$V\in\mathbb{R}^{n\times k}$ 的每一行代表一个任务 v_j 的潜在特征向量 V_j。$\boldsymbol{U}$ 和 $\boldsymbol{V}$ 中的值随机初始化。TimeMF 模型的训练目标是保证 $\boldsymbol{MP}$ 和原始工人任务评分矩阵 $\boldsymbol{MP}$ 的误差最小。目标函数如式(2)所示。

$$\underset{U,V}{argmin}\, d(\boldsymbol{U},\boldsymbol{V})=\frac{1}{2}\sum\nolimits_{i,j}c_{i,j}\left(mp_{ij}-U_iT_j^T\right)^2+\left(\frac{\lambda_U}{2}\sum_{i=1}^{m}\|U_i\|_{Fro}^2+\frac{\lambda_V}{2}\sum_{j=1}^{n}\|V_j\|_{Fro}^2\right) \tag{2}$$

其中，$c_{i,j}$ 用于指示工人 u_i 对任务 v_j 感兴趣的程度，$c_{i,j}=1+\alpha mp_{i,j}$，$\alpha$ 是衰减参数。$\|\cdot\|_{Fro}$ 表示 Frobenius 范数，λ_U 和 λ_V 是正则化项。式(2)可通过梯度下降方法找到最优解。

2.1.3 TimeMF 偏好评分生成

利用上一阶段最终学习到的工人潜在特征向量 $\boldsymbol{U}$ 和任务潜在特征向量 $\boldsymbol{V}$，生成工人对任务的偏好 $\boldsymbol{MP}$。偏好生成的公式如下：

$$mp_{ij}=U_iV_j^T \tag{3}$$

2.2 融入空间因素的双向长短期记忆网络

为了克服工人偏好随工人位置动态改变的问题，使用双向长短期记忆网络(Bidirectional Long Short-Term Memory, BiLSTM)学习工人偏好，模型主要包括三个部分：a)轨迹数据处理。即对轨迹序列中的缺失值填充；b)BiLSTM 预测。通过工人的 GPS 轨迹序列预测工人下一时刻可能会到达的位置；c)BiLSTM 偏好评分生成。基于上一阶段学习到的工人下一时刻可能到达的位置和平台内任务的距离生成工人任务偏好。

2.2.1 轨迹数据处理

对于工人轨迹数据中的缺失值，首先依据工人自身轨迹的时空相关性进行填充，例如，工人 u_i 在 t_s 时刻的轨迹信息缺失，则依次检索该时刻的前 5 个时刻和后 5 个时刻的轨迹信息，若查找到轨迹信息值，则利用该值填充工人 u_i 在 t_s 时刻的轨迹信息 l_s。

然后，对于工人 u_i 的 GPS 轨迹序列仍然存在缺失值的每个时刻，比较工人 u_i 与其他工人之间的轨迹相似性，根据其相似工人在该时刻的轨迹信息来填充轨迹，需要注意的是，如果只有一名相似工人在该时刻有轨迹信息，则直接使用该相似工人的轨迹信息填充，如果有多名工人在该时刻都有轨迹信息，则选择相似性最大的工人所在的轨迹信息填充该时刻。

工人 u_i 和工人 u_w 的轨迹相似性通过下式计算：

$$LS_{iw}=\sum_{t_s\subseteq\mathbb{L}_{u_i}}\xi_{t_s}(l_s^i,l_s^w) \tag{4}$$

其中，$\xi_s(l_s^i,l_s^w)$ 表示 t_s 时刻工人 u_i 和工人 u_w 轨迹位置的相似性，由下式得到：

$$\xi_{t_s}(l_s^i,l_s^w)=\begin{cases}1 &, d(l_s^i,l_s^w)\le\varepsilon\\ 0 &, d(l_s^i,l_s^w)>\varepsilon\end{cases} \tag{5}$$

$d(l_s^i,l_s^w)$ 表示 t_s 时刻工人 u_i 和工人 u_w 的距离，若两个工人之间的位置距离小于等于 ε，认为这两个位置相似，记 $\xi_{t_s}(l_s^i,l_s^w)=1$，否则，$\xi_{t_s}(l_s^i,l_s^w)=0$。ε 的值由平台设定。

2.2.2 BiLSTM 预测

由于 BiLSTM 模型中输入序列长度必须相同，因此在 BiLSTM 模型中引入了滑动窗口方法，时间窗口的大小通过实验指定。工人的轨迹序列首先被转换成固定长度的输入和输出样本集，然后用这些固定长度的样本集来训练 BiLSTM 模型。

BiLSTM 模型总共 5 层：a)输入层；b)嵌入层；c)BiLSTM 层；d)Dropout 层；e)输出层。

a) 输入层：工人任务轨迹序列 $\mathbb{L}_{u_i}=\{(t_1,l_1),(t_2,l_2),\dots,(t_s,l_s),\dots\}$。

b) 嵌入层：通过 Word Embedding 的思想对工人的 GPS 序列信息进行嵌入，得到序列嵌入向量 x_{ID}。

c) BiLSTM 层：包含前向 LSTM 和后向 LSTM 两个隐层。将嵌入向量 x_{ID} 分别输入到两个隐层，得到向量 $h_t=h_t^1\oplus h_t^2$，其中，h_t^1 为前向 LSTM 的输出，h_t^2 为后向 LSTM 的输出。

d) Dropout 层：以预先设定的概率停止神经元的输出，防止模型出现过拟合的问题。

e) 输出层：输出工人下一时刻可能到达的位置 l_{next}。

2.2.3 BiLSTM 偏好评分生成

接下来，建立工人对平台内所有任务的偏好矩阵 LP，LP 中的每一个子元素 lp_{ij} 使用欧氏距离得到。

$$lp_{ij}=\|l_{v_j}-l_{next}\|_2^2 \tag{6}$$

其中，l_{v_j} 表示任务 v_j 要求的感知位置，l_{next} 是工人下一时刻可能到达的位置。如果工人下一时刻可能到达的位置 l_{next} 与任务 v_j 距离很近，则工人很有可能执行该任务。

2.3 偏好融合

将前两个部分的得到的偏好评分 $\boldsymbol{MP}$ 和 $\boldsymbol{LP}$ 使用逻辑回归方法融合，得到工人最终的偏好评分 $\boldsymbol{P}$，$\boldsymbol{P}$ 中的每一个子元素 p_{ij} 通过下式得到：

$$p_{ij}=\frac{1}{1+\exp\left(-\left(mp_{ij}+lp_{ij}\right)\right)} \tag{7}$$

3 任务流行度惩罚因子设计

以往计算任务流行度[21]通过下式计算：

$$pop_j=\frac{\mathbb{F}_{v_j}}{m} \tag{8}$$

$\mathbb{F}_{v_j}$ 是 MCS 平台内执行任务 vj 的工人集合，m 为 MCS 平台总的工人数量。popj 是任务 vj 的流行度值。

然而，这种方法视不同工人对任务流行度的影响是同质的，其结果存在误差。在移动群智感知平台中，任务的流行度与工人声誉有很大关系。一方面，MCS 平台中的工人具有从众心理，工人执行任务的意愿容易收到其他工人的影响，并且，声誉较高的工人在某一时刻执行某个任务的话，则下一时刻容易吸引更多的工人去执行该任务，导致该任务下一时刻的流行度激增。显然，按照以往的做法，即出于覆盖率保证的角度考虑，根据时刻 t 的流行度情况，在 $t+1$ 时刻提升任务的评分，是不合理的。因此，本文引入工人声誉的计算，帮助更好地衡量下一时刻的任务流行情况。

首先，计算工人之间的影响因子 $\boldsymbol{IN}$ 矩阵，即不同工人之间相互影响的难易程度，以工人 u_i 和工人 u_w 为例，二者之间的影响因子 in_{iw}：

$$in_{iw}=\frac{\left|\mathbb{H}_{u_i}\cap\mathbb{H}_{u_w}\right|}{\left|\mathbb{H}_{u_i}\cap\mathbb{H}_{u_w}\right|} \tag{9}$$

其中，$\mathbb{H}_{u_i}$ 和 $\mathbb{H}_{u_w}$ 分别表示工人 u_i 和工人 u_w 执行过的任务集合，若两个工人所做的共同任务数量越多，则表明两个工人的亲密程度越高，两个工人之间也就更容易互相影响。

然后，引入社交网络来计算工人在移动群智感知平台的声誉，建立工人声誉矩阵 $\boldsymbol{WP}$。工人声誉可以通过其他工人的点赞数、评论数量以及工人之间的朋友关系等多种社会关系来度量。

工人 u_i 的声誉对任务 v_j 的流行度影响可以表示为

$$pc_{ij}=wp_i*\sum_{u_w\subseteq\mathbb{U}_{-i}}in_{iw} \tag{10}$$

其中，$\mathbb{U}_{-i}=\{u_1,u_2,\dots u_{i-1},u_{i+1},\dots u_m\}$。因此，任务的流行度重新定

义为

$$pop_j = \frac{\mathbb{F}_{v_j} + \sum_{u_i \subseteq \mathbb{E}_{v_j}} pc_{ij}}{m} \tag{11}$$

接下来，对候选任务集合中的任务流行度进行归一化，使 pop_j 的取值在 0-1 之间。

$$pop_j = \frac{pop_j - pop_{min}}{pop_{max} - pop_{min}} \tag{12}$$

其中，pop_{max} 表示任务集合中所有任务流行度的最大值，pop_{min} 表示任务集合中所有任务流行度的最小值。

接下来，对 MCS 平台中每个任务添加流行度惩罚因子，该流行度惩罚因子可以调节任务的推荐权重，以此保证任务推荐的覆盖率，任务 v_j 的流行度惩罚因子 m_j 由下式计算：

$$m_j = \begin{cases} 1 &, \ pop_j < \phi \text{ and } sample_j < sasmple_j \\ 1 - pop_j &, \ pop_j \geq \phi \text{ and } sample_j < sasmple_j \end{cases} \tag{13}$$

其中，$sample_j$ 为任务 v_j 现在收集到的任务样本数量，$\overline{sample_j}$ 为任务 v_j 总共所需的样本数量，由任务请求方在发布任务请求时自定义确定。ϕ 的取值为平台内所有任务执行次数的中位数。

4　任务推荐列表生成

在任务流行度惩罚因子的基础上，进一步改进工人对任务的预测评分，生成任务推荐列表。工人对任务最终预测评分为

$$y_{ij} = p_{ij} * m_j \tag{14}$$

p_{ij} 为融合时空信息计算出的工人偏好，m_j 为任务 v_j 的流行度惩罚因子。当 $sample_j < \overline{sample_j}$ 时，表明该任务还没有被执行成功，针对每一个工人，依据预测出来的其对任务的评分矩阵 $\boldsymbol{Y}$，将任务按照降序排列，从中选择前 N 个任务组成任务列表推荐给工人，N 的个数由平台自定义设定。若 $sample_j \geq \overline{sample_j}$，表示该任务需要的样本数量已经收集完成，则不再向工人推荐该任务，并将该任务从候选任务集合中删除。

基于时空信息和流行度分析的移动群智感知任务推荐方法的具体实现步骤如下：

算法 1　TimeMF-BiLSTM

输入：工人集合$\mathbb{U}$，任务集合$\mathbb{V}$，每个工人的执行记录 $\mathbb{H}$、任务的执行记录 $\mathbb{F}$、每个工人的轨迹记录$\mathbb{L}$、任务所需样本数量 $\overline{sample_j}$。

输出：工人的任务推荐列表。

a) 工人潜在特征向量 $\boldsymbol{U}$ 和任务潜在特征向量 $\boldsymbol{V}$ 随机初始化。

b) 依据式(1)构建原始工人评分矩阵 $\boldsymbol{MP}$。

c) 构建式(2)的损失函数，通过梯度下降训练 TimeMF 模型；如果损失函数小于阈值，则完成训练，跳到步骤 d)；否则跳到步骤 c)。

d) 依据式(3)构建工人的 TimeMF 偏好评分矩阵 $\boldsymbol{MP}$，$\boldsymbol{MP}$ 中的每一个子元素 mp_{ij} 表示工人 u_i 对任务 v_j 的 TimeMF 偏好。

e) 依据式(5)计算工人 u_i 和工人 u_w 之间的轨迹相似性 $\xi_{t_s}(l_s^i, l_s^w)$。

f) 遍历工人 u_i 的轨迹记录$\mathbb{L}$，如果 $l_s^i = 0$，使用与工人 u_i 轨迹相似性最大的工人在该时隙的位置填充该时隙；否则跳过该时隙，继续遍历轨迹记录$\mathbb{L}$；

g) 训练 BiLSTM 模型，输出工人 u_i 下一时刻可能到达的位置 l_{next}。

h) 依据式(6)计算 l_{next} 和任务 v_j 的相似性 lp_{ij}，构建 BiLSTM 偏好评分矩阵 $\boldsymbol{LP}$，$\boldsymbol{LP}$ 中的每一个子元素 lp_{ij} 表示工人 u_i 对任务 v_j 的 BiLSTM 偏好。

i) 依据式(7)构建工人偏好评分 $\boldsymbol{P}$。$\boldsymbol{P}$ 中的每一个子元素表示工人 u_i 对任务 v_j 的偏好评分。

j) 依据式(13)构建任务流行度惩罚因子矩阵 $\boldsymbol{M}$ 的每一个子元素 m_j 表示任务 v_j 的流行度惩罚因子。

k) 计算工人最终预测评分矩阵 $\boldsymbol{Y}$，$\boldsymbol{Y}$ 中的每一个子元素 y_{ij} 依据式(14)计算。

l) 对于每个工人 u_i，将预测评分矩阵 $\boldsymbol{Y}$ 第 i 行按照 y_{ij} 大小降序排列，取前 N 个作为给工人 u_i 的任务推荐列表输出；如果任务 v_j 收集到的样本数量大于 $\overline{sample}$，则将 v_j 从任务集合$\mathbb{V}$中剔除，转到步骤 a)。

为了更直观地展示 TimeMF-BiLSTM 的任务推荐过程，算法 1 的流程图如图 2 所示。该方法首先训练 TimeMF 模型，得出工人 TimeMF 偏好评分矩阵 $\boldsymbol{MP}$；再对工人轨迹的缺失值填充，输入到 BiLSTM 模型中，预测工人下一时刻可能到达的位置，基于任务与该位置之间的距离得到工人 BiLSTM 偏好评分 $\boldsymbol{LP}$；然后，将 $\boldsymbol{LP}$ 和 $\boldsymbol{MP}$ 的对应元素使用逻辑回归方法融合到一起，综合得到工人的偏好评分 $\boldsymbol{P}$；接下来，构造与任务流行度惩罚因子矩阵 $\boldsymbol{M}$，并将流行度惩罚因子 $\boldsymbol{M}$ 和工人偏好矩阵 $\boldsymbol{P}$ 相乘得到工人对任务的最终评分 $\boldsymbol{Y}$；最后，对应系统中的每个工人 u_i，降序排列工人评分矩阵 $\boldsymbol{Y}$ 的第 i 行，取评分最大的前 N 个任务作为工人 u_i 的任务推荐列表输出；当任务 v_j 收集到的样本数量 $sample_j$ 所需的样本满足要求之后，将该任务从任务集合$\mathbb{V}$中移除，并重新开始模型的训练。

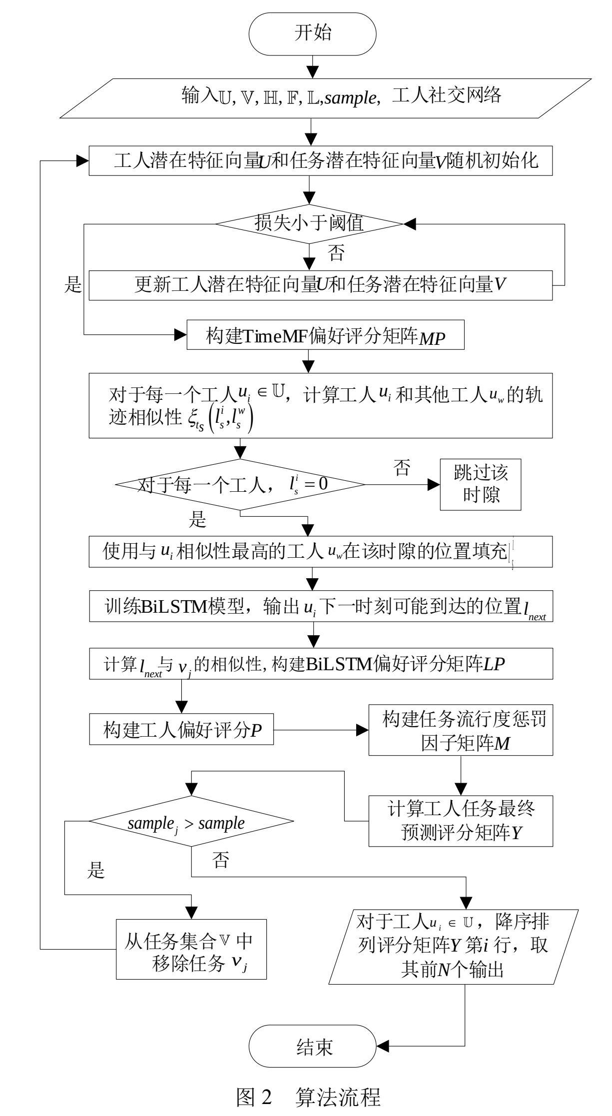


图 2　算法流程

Fig. 2　Algorithm flow chart

5　实验分析

5.1　数据集

在两个真实数据集 Gowalla[23]和 Foursquare[24]上验证所提方法的有效性。所使用数据集的详细信息如下：

a) Gowalla 数据集：该数据集包含了 196591 个用户从 2009 年 8 月至 2010 年 9 月的签到记录。通过处理，比如删除签到记录低于 20 次的工人以及任务地点等，最终提取了 216245 条数据，包括 78 个工人、117 个任务和 371 个位置。

b) Foursquare 数据集：该数据集包含 2153471 个用户，1143092 个场所，1021970 个签到，27098490 个社交关系以

及用户分配给场所的 2809581 评分；所有这些都是通过公共 API 从 Foursquare 应用程序中提取的。在这个数据集中，首先对数据集做预处理操作，比如删除签到记录低于 20 次的工人以及任务地点等。最终剩余 622841 条数据，包括 285 名工人、105 个任务和 168 个地点。

5.2 参数设置

实验运行在一台拥有 Intel Core i5-10210U 处理器，8GB RAM 的戴尔笔记本电脑上。操作系统是 windows 10，编程语言是 Python3.6。所使用的主要参数通过表 1 给出。

表 1 参数设置

Tab. 1 Parameter settings

参数	取值范围	参数	取值范围
时段 T 划分	30min	学习率 α	[0.001,0.05]
任务数 n	[20,100]	正则化参数 β	[0.01,0.9]
工人数 m	15	时间窗口 t	20
潜在特征维度 k	[10,25]	dropout	0.2

5.3 评价准则

a) 准确度(Precision)

$$Precision=\frac{\sum_{1}^{m}\frac{hitcount_i}{total\ recommendation\ number_i}}{m} \tag{15}$$

其中，hitcount$_i$ 是工人从任务推荐列表中执行的任务数量，total recommendation number$_i$ 代表推荐给工人 u_i 的任务数量。

b) 归一化折损累计增益(*NDCG*)

$$NDCG=\frac{DCG}{IDCG} \tag{16}$$

IDCG 是理想条件下的最大 *DCG* 值。*NDCG* 的值介于(0,1)之间，用来衡量任务推荐列表中的任务顺序是否与工人偏好一致。在 MCS 平台中，工人最终选择的任务在任务推荐列表中的位置越高，该任务对折损增益的贡献越大，*NDCG* 的值越大，排名结果越好。

DCG 是折损累计增益，由下式得到：

$$DCG=\sum_{i=1}^{b}\frac{2^{rel_i}-1}{log_2 i+1} \tag{17}$$

其中，b 是任务推荐列表中的任务数量。rel_i 代表该列表中第 i 项的相关度，如果工人执行该项任务，则设置为 1，否则设置为 0。

c) 覆盖率(*Coverage*)

$$Coverage=\frac{1}{n}\sum_{j=1}^{n}(2c-n-1)\,pop(v_c) \tag{18}$$

覆盖率用来描述能够推荐出来得任务占总任务的比例。$pop(v_c)$ 是按照任务流行度 pop 从大到小排序的任务集合中的第 c 个任务。

5.4 对比方法

为了验证 TimeMF-BiLSTM 方法的有效性，本文将其与 6 个基线方法进行比较，用以证明所提出的模型优于其他方法。具体如下所示。

TR-UMCR[8]：该方法使用基于混合用户模型与列表级排序学习算法相结合的协同排序任务推荐方法。

FLTE[11]：该方法利用逻辑回归算法进行任务推荐。

RTRA[14]：该方法仅利用矩阵分解模型学习工人和任务的隐含关系。

PRTR[18]：该方法将工人位置信息和工人偏好结合起来，利用张量分解模型学习工人和任务的关系，忽略工人偏好随时间动态变化的影响。

TCTR[22]：该方法仅融入时间因素考虑工人偏好，忽略工人偏好随位置动态变化的影响。

RLIN：与本文所提出方法 TimeMF-BiLSTM 不同之处是，将两种模型得到的评分以线性加权的形式融合，比较线性加权和逻辑回归偏好融合的不同。

5.5 性能评价

这一节，本文展示与其他基线方法在评估指标上的结果对比。

表 2 和表 3 分别给出了 TimeMF-BiLSTM 任务推荐方法在两个数据集上的精确度效果。从表中可以看出，TimeMF-BiLSTM 的精确度远远高于 *FLTE*，这是由于 *FLTE* 不考虑工人和任务的隐含关系，导致学习到的工人偏好的准确性很差，说明本文模型更符合 MCS 任务推荐的实际应用。当推荐列表长度 N 大于 15 时，能够看到 TimeMF-BiLSTM 的精度大于 *TR-UMCR*、*PRTR* 以及 *TCTR* 算法，说明 TimeMF-BiLSTM 融合时空因素进行偏好融合，可以更加准确地捕捉工人的偏好。可以发现，*TR-UMCR* 和 *PRTR* 算法的效果大致相同，这是由于这两个算法均是采用基本的矩阵分解模型，并且均融入了工人的位置信息来提高推荐效果，这个现象也从侧面反映出，在 MCS 领域，工人执行任务的意愿与工人位置有很大关系。此外，与 *RLIN* 相比，TimeMF-BiLSTM 的精确度略高，这是由于本文使用工人的历史执行情况作为工人对任务的评分，即评分数据都是整数，线性回归方法预测出来的分值必须要通过取整操作，这会造成部分误差。随着推荐列表长度 N 的上升，TimeMF-BiLSTM 的精确度不断上升，在 Gowalla 数据集上，当推荐列表长度 L 为 20 时，TimeMF-BiLSTM 方法的精确度最终收敛到 94.3%左右。综上所述，依据 TimeMF-BiLSTM 方法向工人推荐任务在精确度方面更具有优势。

表 2 不同推荐长度 N 在 Gowalla 数据集的推荐精确度

Tab. 2 Accuracy of different recommended length N on Gowalla dataset

方法	5	10	15	20
TimeMF-BiLSTM	**32.2%**	**59.6%**	**88.4%**	**94.3%**
TR-UMCR	36.8%	58.4%	82.9%	90.2%
FLTE	21.9%	30.7%	64.1%	70.5%
RTRA	25.2%	38.1%	73.8%	82.4%
TCTR	39.6%	59.9%	78.3%	88.9%
PRTR	37.7%	60.3%	80.8%	87.6%
RLIN	30.2%	58.5%	82.1%	91.6%

表 3 不同推荐长度 N 在 Foursquare 数据集上的推荐精确度指标

Tab. 3 Accuracy of different recommended length N on Foursquare dataset

方法	5	10	15	20
TimeMF-BiLSTM	**33.9%**	**57.5%**	**78.9%**	**92.1%**
TR-UMCR	27.7%	57.1%	77.5%	89.2%
FLTE	21.0%	25.1%	57.1%	63.3%
RTRA	25.2%	33.9%	70.1%	78.5%
TCTR	20.7%	43.5%	77.4%	89.2%
PRTR	37.7%	60.3%	84.8%	85.1%
RLIN	31.6%	53.4%	81.8%	89.5%

另外，从表 2 和表 3 可以发现，当推荐列表长度低于 10 时，TimeMF-BiLSTM 的精度略小于 *PRTR*，这是因为本文的算法考虑了任务的流行度惩罚，导致推荐的精度稍有降低。为进一步验证该现象，本文从 Foursquare 数据集随机抽取特定用户以及其历史执行任务的记录等相关信息作为测试样本。具体来说，本文选择的工人 ID 号为 9 号，取任务集合中 ID 从 1 至 20 号任务作为训练集，工人 9 对这 20 个任务的真实偏好评分值以及依据本文所提方法对这些任务的预测评分情况如表 4 所示。从表 4 可以看出，理论上来讲，依据工人对不同任务的评分高低来生成任务列表，工人 9 真实最优的任务 ID 列表应该为*[3,4,8,2,6,7,1,5,9,10]*。

然后，依据本文所提方法预测工人 9 对这 10 项任务的

评分，结果显示，所提方法对工人 9 的预测偏好评分 $\boldsymbol{P}$ 结果在表 4 展示，可以发现预测偏好评分 P 与工人 9 的真实评分的均方误差为 0.414152，与工人 9 的真实偏好非常接近，证明本文所提方法在预测工人偏好方面的准确性。但是，如果仅以此作为推荐结果的话，任务 2、3、4、8 由于较高的评分，工人 9 下一次的推荐列表前 5 个位置将全是任务区域 C_3 和 C_4 的任务，这会导致工人 9 越来越偏爱于任务区域 C_3 和 C_4 的任务，那么任务区域 C_3 和 C_4 的任务流行度就会越来越高，而工人 9 执行任务的总次数是一定的，那么相应地，任务区域 C_1 和 C_2 则会由于缺乏足够的推荐而收集不到充分的样本。因此，本文使用流行度惩罚因子，调整流行度过高的任务权重，得到最终的工人任务预测评分矩阵 $\boldsymbol{Y}$，结果在表 4 展示，可以发现调整之后工人 9 的最终预测评分 $\boldsymbol{Y}$ 与工人 9 的真实评分的均方误差稍有提升，但是任务 2、3、4、8 的推荐权重明显降低，任务区域 C_1 和 C_2 区域处于工人下一时刻的任务列表中靠前位置，这对于提升任务推荐的覆盖率是非常有帮助的，这一现象与本文的初衷，也就是适当降低推荐精度，促进推荐效果覆盖率，是非常一致的。而随着推荐列表长度不断增加，也就是说一次性推荐给工人的任务数量不断增加，那么每一次推荐的任务列表所能覆盖的范围就会越来越来，即推荐效果的覆盖率越来越大，此时，为满足覆盖率要求而设置的流行任务的惩罚因子就会越来越小，那么推荐效果的精确度也就会逐渐增加了。

表 4 样本实例

Tab. 4 Sample instance

任务 ID	真实评分	任务所属区域	预测偏好评分 $\boldsymbol{P}$	最终预测评分 $\boldsymbol{Y}$
1	2	C_1	2.78933511	2.78933511
2	3	C_4	3.56075737	1.61037869
3	6	C_3	5.54626787	0.00000000
4	5	C_4	4.39667412	0.92437798
5	2	C_2	3.09950478	3.09950478
6	3	C_4	3.09983137	3.09983137
7	3	C_2	2.95584221	2.95584221
8	5	C_3	4.54064423	0.75677404
9	1	C_1	1.60706052	1.60706052
10	0	C_1	0.63975154	1.63975154

接下来，将推荐长度 N 设置为 20，图 3 展示 TimeMF-BiLSTM 与其他基线方法在 $NGDD$ 指标的性能对比。本文将计算折损累计增益时的参数 b 设置为 $\boldsymbol{n}$，$\boldsymbol{n}$ 是 MCS 平台中的任务数量，由平台当前的候选任务数量决定。可以发现，TimeMF-BiLSTM 的 $NDCG$ 明显高于其他基线方法，并且比逻辑回归方法的 $NDCG$ 高了将近一倍左右，证明本文的模型评价排序的准确性。

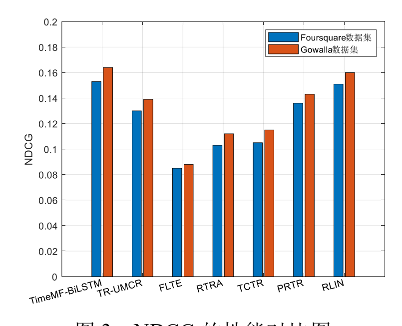


图 3 NDCG 的性能对比图

Fig. 3 Comparison of NDCG

图 4 和图 5 显示不同方法在两个数据集上的覆盖率指标情况，本文将 TimeMF-BiLSTM 方法及其他基线算法重复运行 50 次，取其平均值作为不同算法的覆盖率值。从图中可以看出，覆盖率与推荐列表的长度 N 呈现正相关关系，并且相较于其他基线方法，本文算法具有更高的覆盖率。其原因是，以往的算法没有考虑任务的流行度偏差，造成过度推荐流行性任务。本文利用任务流行度设立惩罚因子，进而降低流行性过高的任务权重，提高冷门任务选择的概率，因此当任务数量不断增加时，TimeMF-BiLSTM 任务推荐方法也能保证工人有更多的机会选择其他任务。

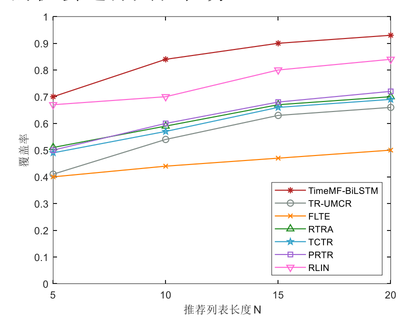


图 4 覆盖率性能对比(Gowalla 数据集)

Fig. 4 Comparison of coverage (Gowalla dataset)

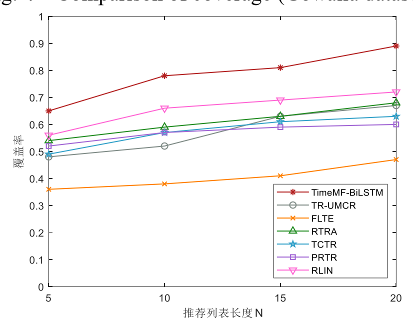


图 5 覆盖率性能对比(Foursquare 数据集)

Fig. 5 Comparison of coverage (Foursquare dataset)

最后，在两个数据集上进行实验，以验证工人声誉关系对任务覆盖率的影响，如图 6 和图 7 所示。推荐列表长度为 20，图 5 中的不同线条表明本文计算工人声誉时引入的社交网络信息条数，横坐标表示移动群智感知平台中工人和任务的比率，理论上来说，当平台内工人和任务的比率不断增加时，推荐的覆盖率会成比例下降。从图中可以看出，工人声誉的引入，可以保证任务推荐覆盖率的下降速度变缓慢，这是由于工人声誉可以让本文更准确的分析任务流行度，对流行任务的惩罚也就更加准确，因此当工人和任务的比率不断增加时，TimeMF-BiLSTM 方法也能保证任务推荐的覆盖率下降速度最慢。从图中可以看出，当任务和工人的比率为 10 时，平台的覆盖率仍可以超过 71%。并且从图中可以发现，随着工人社会关系引入条数的不断增加，覆盖率下降速度越来越慢，因此在保证任推荐的覆盖率时有必要引入工人之间的社会关系。这些实验结果表明，TimeMF-BiLSTM 在 MCS 实际应用中具有更全面的性能。

6 结束语

本文针对移动群智感知领域在挖掘工人偏好时忽略时空信息和任务流行度影响，导致任务推荐的准确性和覆盖率低的问题，设计了融合时空信息和任务流行度分析的任务推荐方法。融合时空信息可以准确地挖掘工人偏好，而任务流行度分析可以有效地缓解现有研究任务覆盖率低的问题，所提方法将时间因素、空间因素两个方面对推荐的影响分开计算，可以有效地缓解现有方法需要频繁更新的缺点。本文所提方法适用于所有具有时空背景的项目推荐问题，同时由于不同模型相互独立，未来更多影响工人意愿的因素可以轻松嵌入，并且所提方法将推荐的精度和覆盖率两个指标分开优化求解，支持其他多种优化目标组合，具有良好的可扩展性。在两个真实数据集上实验本文方法的有效性，实验结果表明，与

有的基线方法相比，本文所提方法在准确率和覆盖率具有良好的效果，证明了该方法在处理移动群智感知任务推荐领域实际应用方面具有出色的性能。在今后的工作中，本文将考虑更多可能影响 MCS 平台任务推荐效果的因素，并进一步研究如何用更短的训练时长来及时捕捉工人偏好，为工人提供更好的服务体验。

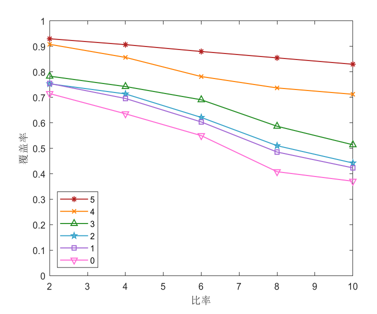


图 6 工人声誉对两个数据集覆盖率的影响(Gowalla 数据集)

Fig. 6　Influence of worker reputation on accuracy (Gowalla dataset)

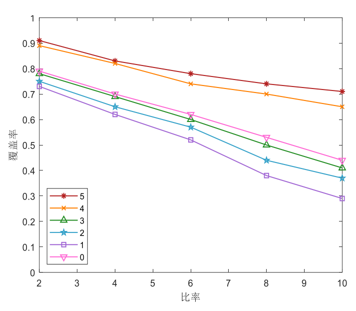


图 7　工人声誉对两个数据集覆盖率的影响(Foursquare 数据集)

Fig. 7　Influence of worker reputation on accuracy (Foursquare dataset)

参考文献：

[1] Ganti R K, Ye F, Lei H. Mobile crowdsensing: current state and future challenges [J]. IEEE Communication Magazine, 2011, 49 (11): 32-39.

[2] Montori F, Bedogni L, Bononi L. A collaborative internet of things architecture for smart cities and environmental monitoring [J]. IEEE Internet of Things Journal, 2018, 5 (2): 592-605.

[3] Jovanović S, Jovanović M, Škorić T, *et al.* A mobile crowd sensing application for hypertensive patients [J]. Sensors. 2019, 19 (2): 400-416.

[4] Huang Junqin, Kong Linghe, Dai Hongning, *et al.* Blockchain based mobile crowd sensing in industrial systems [J]. IEEE Trans on Industrial Informatics, 2019, 16 (10): 6553-6563.

[5] Kim K, Zabihi H, Kim H, *et al.* TrailSense: a crowdsensing system for detecting risky mountain trail segments with walking pattern analysis [J]. Proceedings of the Acm on Interactive Mobile Wearable & Ubiquitous Technologies, 2017, 1 (3): 1-31.

[6] Wan Jiafu, Liu Jianqi, Shao Zehui, *et al.* Mobile crowd sensing for traffic prediction in internet of vehicles [J]. Sensors, 2016, 16 (1): 88-102.

[7] Yucel F, Yuksel M, Bulut E. QoS-based budget constrained stable task assignment in mobile crowdsensing [J]. IEEE Trans on Mobile Computing, 2021, 20 (11): 3194-3210.

[8] 王健，刘嘉欣，赵国生，等. 移动群智感知中基于协同排序的任务推荐方法 [J]. 电子学报，2021, 49 (10): 2012-2019. (Wang Jian, Liu Jiaxin, Zhao Guosheng, et at. Task recommendation method based on collaborative ranking in mobile crowd sensing [J]. Acta Electronica Sinica, 2021, 49 (10): 2012-2019.)

[9] Zhao Zhongnan, Wang Yanli, Wang Huiqiang. SDN-based cross-domain cooperative method for trusted nodes recommendation in Mobile crowd sensing [J]. Peer-to-Peer Networking and Applications, 2021, 14 (5): 3793–3805.

[10] Tang Wenjuan, Zhang Kuan, Ren Ju, *et al.* Privacy-preserving task recommendation with win-win incentives for mobile crowdsourcing [J]. Information Sciences, 2020, 527: 477-492.

[11] Karaliopoulos M, Koutsopoulos I, Titsias M. First learn then earn: optimizing mobile crowdsensing campaigns through data-driven user profiling [C]// Proc the 17th ACM International Symposium on Mobile Ad Hoc Networking and Computing. New York: ACM Press, 2016: 271–280.

[12] Wang Zhibo, Zhao Jing, Hu Jiahui, *et al.* Towards personalized task-oriented worker recruitment in mobile crowdsensing [J]. IEEE Trans on Mobile Computing, 2021, 20 (5): 2080-2093.

[13] Kumar P, Thakur R S. Recommendation system techniques and related issues: a survey [J]. International Journal of Information Technology, 2018, 10 (4): 495–501.

[14] Yuen M C, King I, Leung K S. Taskrec: a task recommendation framework in crowdsourcing systems [J]. Neural Processing Letters, 2015, 41 (2): 223-238.

[15] Wang Jian, Liu Jiaxin, Zhao Zhongnan, *et al.* A task recommendation framework for heterogeneous mobile crowdsensing [J]. The Journal of Supercomputing, 2021, 2021 (48): 12121–12142.

[16] Dai Wei, Wang Yufeng, Ma Jianhua, *et al.* BTR: a feature-based bayesian task recommendation scheme for crowdsourcing system [J]. IEEE Trans on Computational Social Systems, 2020, 7 (3): 780-789.

[17] Yang Shuo, Zheng Zhenzhe, Tang Shaojie, *et al.* Fine-grained user profiling for personalized task matching in mobile crowdsensing [J]. IEEE Trans on Mobile Computing, 2021, 20 (10): 2961-2976.

[18] Yang Guisong, Li Yanting, He Xingyu, *et al.* Profile-free and real-time task recommendation in mobile crowdsensing [J]. IEEE Trans on Computational Social Systems, 2021, 8 (6): 1311-1322.

[19] Noulas A, Scellato S, Lathia N, *et al.* Mining user mobility features for next place prediction in location-based services [C]// IEEE International Conference on Data Mining. Piscataway, NJ: IEEE Press, 2012: 1038-1043.

[20] Song Shiwei, Liu Zhidan, Li Zhenjiang, et at. Coverage-oriented task assignment for mobile crowdsensing [J]. IEEE Internet of Things Journal, 2020, 7 (8): 7407-7418.

[21] 王宁，张巍，苏湛，等. 推荐系统中考虑流行程度差异的评分预测算法 [J]. 软件导刊，2021, 20 (11): 18-21. (Wang Ning, Zhang Wei, Su Zhan, et at. Rating prediction algorithm considering popularity differences in recommender systems [J]. Software Guide, 2021, 20 (11): 18-21.)

[22] Mcy A, Ik B, Ksl B. Temporal context-aware task recommendation in crowdsourcing systems [J]. Knowledge-Based Systems, 2021, 219 (1): 106770.

[23] Scellato S, Noulas A, Lambiotte R, *et al.* Socio-spatial properties of online location-based social networks [C]// Proc of the 5th International Conference on Weblogs and Social Media. Barcelona: [s. n.] , 2011: 329-336.

[24] Vasconcelos M A, Ricci S, Almeida J, *et al.* Tips, dones and todos: uncovering user profiles in foursquare [C]// Proc of the 5th ACM International Conference on Web Search and Data Mining. New York: ACM Press, 2012: 653-662.